アルゴリズムとデータ構造 理解度確認レポート（前半）

学籍番号:62115799
氏名: 平井優我

[基礎問題 01-01]

まず、$4x^2 - 9 = 0$ の方程式の解は、

$$x = \pm\frac{3}{2} \tag{1}$$

となる。ニュートン法によってこの解のどちらかが得られることを示す。整数 $n \geq 0$ を用いて、n 番目の x 軸との切片を x_n とおく。このとき解析的に以下のような漸化式が求まる (ただし、初期値が 0 のときはニュートン法を用いても切片がもとまらないことは自明)。

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{9}{4}\frac{1}{x_n}\right)\ ,\ (x_n \neq 0) \tag{2}$$

今、$x_n > 0$ のとき $x_{n+1} > 0$、$x_n < 0$ のとき $x_{n+1} < 0$ となる。また、$x_n > 0$ だとして (2) 式の両辺に -1 をかけると、その漸化式は $x_n < 0$ としたときと同じ式を表す。これより、$x_n > 0$ だとして得られた議論について、x_n の符号を逆にすれば $x_n < 0$ のときの議論が再現できることがわかる。よって、これ以降は $x_n > 0$ の範囲のみで議論する。

まず、x_n と x_{n+1} の符号は一致することから、必ず $x_n \neq 0$ である。(2) 式の x_n, x_{n+1} をすべて α に置き換えて x について解くと、

$$\alpha = \frac{1}{2}\left(\alpha + \frac{9}{4}\frac{1}{\alpha}\right) \tag{3}$$

$$\therefore\ \alpha = \frac{3}{2}\ \ (x_n < 0\text{のときは負の解をとる。}) \tag{4}$$

と求まる。(2) 式から (3) 式を引くと、

$$x_{n+1} - \alpha = \frac{1}{2}(x_n - \alpha) + \frac{9}{8}\left(\frac{1}{x_n} - \frac{1}{\alpha}\right) \tag{5}$$

となる。ここで、重要となるのは (5) 式右辺の第 2 項の符号である。$x_{n+1} > 0$、$x_n > 0$ であることから、(2) 式について相加・相乗平均を用いると、

$$\begin{aligned} x_{n+1} &= \frac{1}{2}\left(x_n + \frac{9}{4}\frac{1}{x_n}\right) \\ &\geq \sqrt{\frac{9}{4}} = \frac{3}{2} \end{aligned} \tag{6}$$

となる。これは、どのような初期値を持ってきたとしても x_1 以降は必ず 3/2 以上になることを表している。よって、$n \geq 1$ について、(5) 式右辺の第 2 項の符号は負となり、

$$|x_{n+1} - \alpha| \leq \frac{1}{2}|x_n - \alpha| \tag{7}$$

が成り立つ。(7) 式を繰り返し用いることによって、

$$|x_n - \alpha| \leq \frac{1}{2^{n-1}}|x_1 - \alpha| \tag{8}$$

となる。よって、$n \to \infty$ としてはさみうちの原理から、

$$\lim_{n\to\infty} x_n = \alpha = \frac{3}{2} \tag{9}$$

となる。これは、(1) で求めた正の解と一致する。

以上をまとめると、

初期値	求まる解
正	$\frac{3}{2}$
負	$-\frac{3}{2}$
0	求まらない

となる。

[基礎問題 02-01]

2 進数表示したい 10 進数の数を b とすると、b を 2 で n 回割ったときの商を b_n、余りを d_n とする。b を 2 進数表示するということは、

$$b = a_0 2^0 + a_1 2^1 + a_2 2^2 + \cdots \tag{10}$$

というふうに表すことと同値である。b, b_n を 2 で割ったときの商と余りで表すと、

$$\begin{aligned} b &= 2b_0 + d_0 \\ b_0 &= 2b_1 + d_1 \\ b_1 &= 2b_2 + d_2 \\ &\vdots \\ b_{n-2} &= 2b_{n-1} + d_{n-1} \qquad (11) \\ b_{n-1} &= 2 \cdot 0 + d_n \qquad (12) \end{aligned}$$

となる。ただし $b_n = 0$ となるまで繰り返す。(12) 式を 1 つ前の (11) 式へ代入し、得られた式を次々とその 1 つ前の式に代入していくと、

$$b = d_0 2^0 + d_1 2^1 + d_2 2^2 + \cdots + d_n 2^n \tag{13}$$

と表せることがわかる。よって、$a_i = d_i$ とすることによって、b を 2 進数で表すことができる。

次の表に 20231201 について、d_i を示す。

d_1	1	d_{11}	1	d_{21}	1
d_2	0	d_{12}	0	d_{22}	1
d_3	0	d_{13}	1	d_{23}	0
d_4	0	d_{14}	1	d_{24}	0
d_5	0	d_{15}	0	d_{24}	1
d_6	1	d_{16}	1		
d_7	0	d_{17}	0		
d_8	0	d_{18}	0		
d_9	0	d_{19}	1		
d_10	0	d_{20}	0		

よって、

$$20231201 = 1001101001011010000100001_{(2)} \tag{14}$$

となる。

[基礎問題 02-02]

例えば $a_7a_6a_5a_4a_3a_2a_1a_{0(2)}$ とそれぞれの位の数が a_n で表されているような数を考える。次のような式変形を行うことによって、

$$\begin{aligned} a_7a_6a_5a_4a_3a_2a_1a_{0(2)} &= a_7a_6a_5a_4 0000_{(2)} + a_3a_2a_1a_{0(2)} \\ &= a_7a_6a_5a_{4(2)} \times 10000_{(2)} + a_3a_2a_1a_{0(2)} \end{aligned} \tag{15}$$

と表せる。ここでそれぞれの数値を $a_7a_6a_5a_{4(2)} = A_{1(16)}, a_3a_2a_1a_{0(2)} = A_{0(16)}$ として 16 進数で表すと、

$$\begin{aligned} a_7a_6a_5a_4a_3a_2a_1a_{0(2)} &= A_{1(16)} \times 10 + A_{0(16)} \\ &= A_1A_{0(16)} \end{aligned} \tag{16}$$

となる。これは、2 進数は 1 の位から 4 桁ずつに区切って 16 進数にすることができることを表している。

上記のことを用いると、

$$\begin{aligned} 1000110111000001111000111_{(2)} &= 11\mathrm{B}83\mathrm{C}7_{(16)} \\ &= 16^6 + 16^5 + 11 \cdot 16^4 + 8 \cdot 16^3 + 3 \cdot 16^2 + 16 \cdot 12 + 7 \\ &= 18580423 \end{aligned} \tag{17}$$

となる。

[基礎問題 02-03]

A	B	X
0	0	1
1	0	1
0	1	1
1	1	1

[基礎問題 03-01]

(i) 3612 は 3 の倍数かつ 5 の倍数でないから、Fizz と出力される。

(ii) 30330 は 3 の倍数かつ 5 の倍数であるから、FizzBuzz と出力される。

[基礎問題 03-02]

(i) $T_1(N) = N^2 + 2N\log N + 3N\sqrt{N} = \mathcal{O}(N^2)$

(ii) $T_1(N) = 2^N + N^{2023} = \mathcal{O}(2^N)$

[基礎問題 04-01]

$$
\begin{aligned}
f(3) &= 4 \times 3^2 - 3 \times 3 + 2 \times f(1) + 5 \times f(2) \\
&= 82 \\
f(4) &= 4 \times 4^2 - 3 \times 4 + 2 \times f(2) + 5 \times f(3) \\
&= 480 \\
f(5) &= 4 \times 5^2 - 3 \times 5 + 2 \times f(3) + 5 \times f(4) \\
&= 2649 \\
f(6) &= 4 \times 6^2 - 3 \times 6 + 2 \times f(4) + 5 \times f(5) \\
&= 14331
\end{aligned}
$$

プログラム

```
f=[5,9]
n=4
for i in range(n):
  f.append(4*((i+3)**2)-3*(i+3)+2*f[i]+5*f[i+1])
print(f)
```

実行結果

```
[5, 9, 82, 480, 2649, 14331]
```

[基礎問題 04-02]

$$
\begin{aligned}
f(2) &= 5 \times 2 + g(1) \\
&= 12 \\
g(2) &= 2 \times 2 \times f(1) \\
&= 28 \\
f(3) &= 5 \times 3 + g(2) \\
&= 43 \\
g(3) &= 2 \times 3 \times f(2) \\
&= 72 \\
f(4) &= 5 \times 4 + g(3) \\
&= 92 \\
g(4) &= 2 \times 4 \times f(3) \\
&= 344 \\
f(5) &= 5 \times 5 + g(4) \\
&= 369 \\
g(5) &= 2 \times 5 \times f(4) \\
&= 920
\end{aligned}
$$

プログラム

```
f=[7]
g=[2]
n=4
for i in range(n):
  f.append(5*(i+2)+g[i])
  g.append(2*(i+2)*f[i])
print(f,g)
```

実行結果

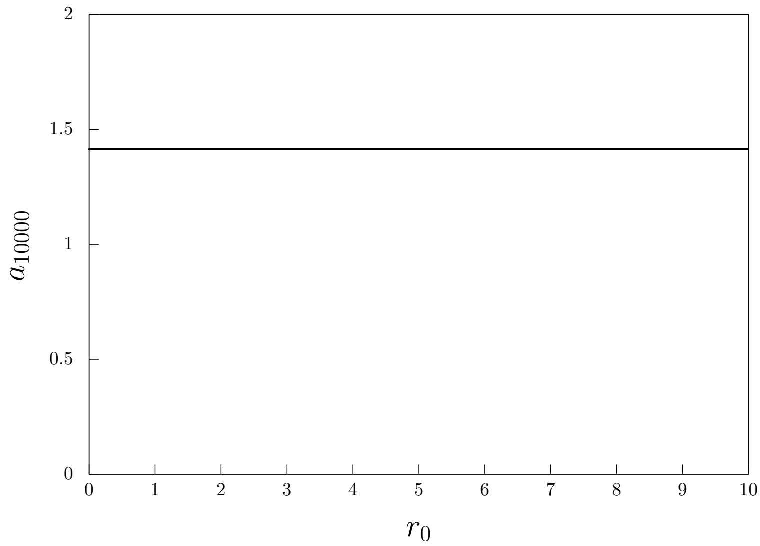
```
[7, 12, 43, 92, 369] [2, 28, 72, 344, 920]
```

[基礎問題 04-03]

Python を用いて再帰関数コードを作成すると、

```
a=int(input())
n=int(input())
for i in range(n-1):
    a=1+1/(1+a)
print(a)
```

となる。r_0 と a_{10000} の関係を下図に示す。



よって、正の実数 r_0 について、a_n は $\sqrt{2}$ に収束する。

[基礎問題 04-04]

Python を用いて再帰関数コードを作成すると、

```
F=[0,0,0,1]
n=int(input())
for i in range(n-3):
  F.append(F[i]+F[i+1]+F[i+2]+F[i+3])
print(F[n])
```

となる。これより、$F_{15} = 1490, F_{38} = 5350220959$ と求まる。

[基礎問題 05-01]

ソートアルゴリズム名	平均計算量
選択ソート	$\mathcal{O}(n^2)$
バブルソート	$\mathcal{O}(n^2)$
挿入ソート	$\mathcal{O}(n^2)$
マージソート	$\mathcal{O}(n \log n)$
クイックソート	$\mathcal{O}(n \log n)$

選択ソートでは要素が n 個あるとき、以下のような計算を行う。

一番左の要素と右側 $n-1$ 個の大小を比較し交換する。

$\downarrow$

一番左の要素と右側 $n-2$ 個の大小を比較し交換する。

$\downarrow$

$\vdots$

$\downarrow$

一番左の要素と右側 1 個の大小を比較し交換する。

よって、計算量 $T(n)$ は、

$$
\begin{aligned}
T(n) &= (n-1) + (n-2) + \cdots + 2 + 1 \\
&= \frac{1}{2}n(n-1) \\
&= \mathcal{O}(n^2)
\end{aligned}
$$

となる。

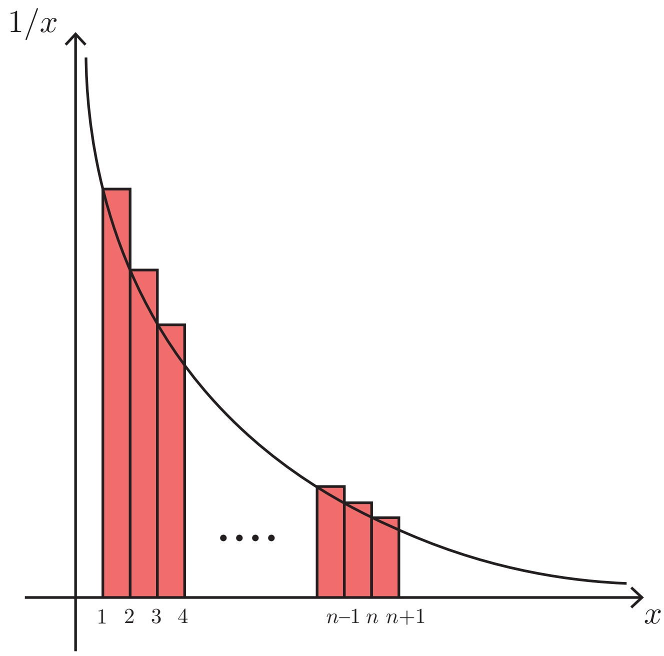
[基礎問題 06-01]

(1) 8,9,10

(2) 5

(3) 4,6,7,9,11,12,13,14,15

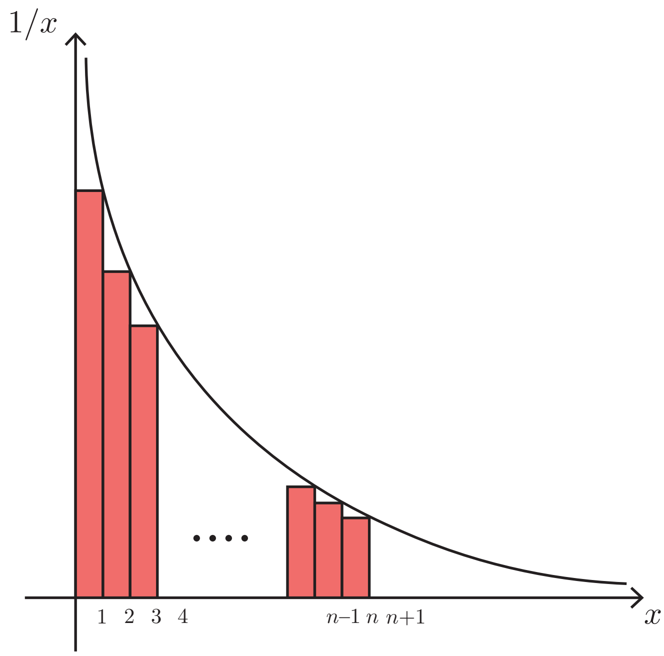
[基礎問題 06-02]



上図のような求積を考えると、

$$\sum_{k=1}^{n} \frac{1}{k} > \int_{1}^{n+1} \frac{1}{x} \mathrm{d}x = \log{(n+1)} = \mathcal{O}(\log n) \tag{18}$$

となる。



次に上図のような求積を考えると、

$$\sum_{k=1}^{n} \frac{1}{k} < 1 + \int_{1}^{n} \frac{1}{x} \mathrm{d}x = 1 + \log{(n)} = \mathcal{O}(\log n) \tag{19}$$

となる。よって、(18) 式と (19) 式から、

$$\sum_{k=1}^{n} \frac{1}{k} = \mathcal{O}(\log n) \tag{20}$$

となる。

[発展問題 03-01]

　a, b を $a > b > 1$ となる整数とし、割り算

$$\begin{aligned} a &= bq_1 + r_1,\ 0 < r_1 < b \\ b &= r_1 q_2 + r_2,\ 0 < r_2 < r_1 \\ &\vdots \\ r_{l-2} &= r_{l-1} q_l + r_l,\ r_{l-1} > r_l = 0 \end{aligned}$$

により整数 l と $q_1, \cdots, q_l, r_1, \cdots, r_l$ を定める。また、数列 $\{F_n\}$ をフィボナッチ数列とし、α を $x^2 = x + 1$ の解とする。

　まず、① $b \geq F_{l+1}$ を示す。

$$\begin{aligned} b &\geq r_1 + r_2 \\ &\vdots \\ r_{k-2} &\geq r_{k-1} + r_k\ (2 < k < l) \\ &\vdots \\ r_{l-2} &> r_{l-1} > r_l = 0 \end{aligned}$$

が成り立つから、$r_{l-1} = 1, r_{l-2} = 2, \cdots, r_{k-2} = r_{k-1} + r_k, \cdots, b = r_1 + r_2$ のとき、つまり $b = F_{l+1}$ のとき b は最小となる。よって、$b \geq F_{l+1}$ である。

　次に② $F_{n+1} \geq \alpha^{n-1}$ を数学的帰納法によって示す。

(i) $n = 1, 2$ のとき、

$$F_2 = 1, F_3 = 2 > \frac{1+\sqrt{5}}{2} = \alpha$$

が成り立つ。

(ii) $n = k, k+1$ (k : 正の整数) のとき $F_{n+1} \geq \alpha^{n-1}$ が成り立つとする。このとき、$\alpha^2 = \alpha + 1$ であるから、

$$\begin{aligned} F_{k+3} &= F_{k+1} + F_{k+2} \\ &\geq \alpha^{k-1} + \alpha^k \\ &= \alpha^{k-1}(1+\alpha) \\ &= \alpha^{k-1}\alpha^2 \\ &= \alpha^{k+1} \end{aligned}$$

となり、$n = k+2$ のとき、$F_{n+1} \geq \alpha^{n-1}$ が成り立つ。

(i)、(ii) から、すべての正の整数 n に対して $F_{n+1} \geq \alpha^{n-1}$ が成り立つ。

　最後に③ $\alpha^5 > 10$ であることを示す。$\alpha = (1+\sqrt{5})/2$ であるから、$(1+\sqrt{5})^5 > 10 \cdot 2^5 = 320$ を示せばよい。また、$(1+\sqrt{5})^5 = 176 + 80\sqrt{5}$ より、$80\sqrt{5} > 144$ を示すことと等価となる。よって、$\left(80\sqrt{5}\right)^2 = 32000 > 20736 = 144^2$ より、$\alpha^5 > 10$ が成り立つ。

　これらの公理から題意は示される。2 つの整数 x, y の最大公約数を (x, y) で表すと、

$$(a, b) = (b, r_1) = \cdots = (r_{l-2}, r_{l-1}) = (r_{l-1} q_l, r_{l-1}) = r_{l-1}$$

であるから、$l \geq 5d$ を示せばよい。$b < 10^d$ と①、②を合わせると、

$$\alpha^{l-1} \leq F_{l+1} \leq b < 10^d$$

が得られる。これと③を合わせると、

$$10^{l-1} < \alpha^{5(l-1)} < 10^{5d}$$

となるから、$l-1 < 5d$ つまり $l < 5d+1$ であり、$l \leq 5d$ が成り立つ。

[発展問題 03-02]

$a = r_0, b = r_1$ と置く。このとき、$\mathrm{GCD}(a,b) = r = r_n$ は次の式で与えられる。

$$r_0 = q_1 r_1 + r_2,\ 0 < r_2 < r_1 \tag{21}$$
$$r_1 = q_2 r_2 + r_3,\ 0 < r_3 < r_2 \tag{22}$$
$$r_2 = q_3 r_3 + r_4,\ 0 < r_4 < r_3 \tag{23}$$
$$\vdots$$
$$r_{k-1} = q_k r_k + r_{k+1},\ 0 < r_{k+1} < r_k \tag{24}$$
$$\vdots$$
$$r_{n-2} = q_{n-1} r_{n-1} + r_n,\ 0 < r_n < r_{n-1} \tag{25}$$
$$r_{n-1} = q_n r_n \tag{26}$$

このとき $i = 0, 1, 2, \cdots, k$ に対して、

$$ax_i + by_i = r_i \tag{27}$$

と表されたとすると、

$$x_{k+1} = x_{k-1} - q_k x_k \tag{28}$$
$$y_{k+1} = y_{k-1} - q_k y_k \tag{29}$$

対して、

$$ax_{k+1} + by_{k+1} = r_{k+1} \tag{30}$$

が成立する。

このことを証明する。まず、(24) 式から

$$r_{k+1} = r_{k-1} - q_k r_k$$

となるが、この右辺に (27) 式の $i = k-1$ と $i = k$ のときを代入すると、

$$\begin{aligned} r_{k+1} &= (ax_{k-1} + by_{k-1}) - q_k (ax_k + by_k) \\ &= a(x_{k-1} - q_k x_k) + b(y_{k-1} - q_k y_k) \end{aligned}$$

この式に (28)、(29) 式を代入すると、

$$r_{k+1} = x_{k+1} a + y_{k+1} b$$

が得られる。

ここで、$x_0 = 1, y_0 = 0, x_1 = 0, y_1 = 1$ と置く。このとき、$a = r_0, b = r_1$ と (21)、(22) 式に注意すると、$i = 0, 1$ に対して、

$$ax_i + by_i = r_i$$

が成立する。したがって、(27)、(28)、(29)、(30) 式の操作を $k = 2, 3, \cdots, n-1$ まで続けると、

$$ax_n + by_n = r_n$$

が得られる。ここで、$r_n = \mathrm{GCD}(a,b) = c$ であるから、$x = x_n, y = y_n$ と置くと、

$$ax + by = c$$

が得られる。

$a=9876, b=5432$ とき、ユークリッドの互除法から、

$$\begin{aligned}9876 &= 5432\times 1+4444\\ 5432 &= 4444\times 1+988\\ 4444 &= 988\times 4+492\\ 988 &= 492\times 2+4\\ 492 &= 4\times 123\end{aligned}$$

となり、$\mathrm{GCD}(a,b)=4$ とわかる。またこれより、数列 $\{q_n\}\{x_n\}\{y_n\}$ を求めると、

$$\begin{aligned}\{q_n\} &= \{1,1,4,2,123\}\\ \{x_n\} &= \{1,0,1,-1,5,-11\}\\ \{y_n\} &= \{0,1,-1,2,-9,20\}\end{aligned}$$

とわかる。これにより、求める x,y は、

$$(x,y)=(-11,20)$$

である。

[発展問題 04-01]

Python によってコードを作成すると次のようになる。

```
cache={}
def tarai(x,y,z):
  if x<=y:
    return y
  key=(x,y,z)
  val=cache.get(key)
  if val != None:
    return val
  val=tarai(tarai(x-1,y,z),tarai(y-1,z,x),tarai(z-1,x,y))
  cache[key]=val
  return val
```

用いた工夫として、一度計算した $\mathrm{Tarai}(x,y,z)$ については記憶しておき、2 度と計算しないようにする点である。これを実装するために Python で辞書を用いた。

これにより、

$$\mathrm{Tarai}(20,10,0)=20$$

となる。

[発展問題 06-01]

$$T(n)\leq\frac{1}{n}\sum_{i=1}^{n}\{T(i-1)+T(n-i)+c_1(n-1)\},\ (c_1>0)$$

は、$\sum$ の中身を計算することで、

$$T(n)\leq\frac{2}{n}\sum_{i=0}^{n-1}T(i)+c_1(n-1) \tag{31}$$

ここで、

$$T(n) \leq 2n\ln n - \frac{c_1}{2}(n-1) \tag{32}$$

が成り立つことを帰納法を用いて示す。

(i) $n=1$ のとき、$T(1)=0$ で等号が成り立つことは自明である。

(ii) k 以下の全ての正の整数 n (kは正の整数) について、(32) 式が成り立つと仮定すると、$n=k+1$ のとき、

$$\begin{aligned}
T(k+1) &\leq \frac{2}{k+1}\sum_{i=0}^{k} T(i) + c_1 k \\
&\leq \frac{2}{k+1}\sum_{i=1}^{k} \left\{2i\ln i - \frac{c_1}{2}(i-1)\right\} + c_1 k + \frac{2}{k+1}T(0) \\
&\leq \frac{4}{k+1}\int_1^{k+1} \left\{x\ln x - \frac{c_1}{2}(x-1)\right\} \mathrm{d}x + c_1 k,\ (T(0)=0) \\
&= \frac{4}{k+1}\left[\frac{x^2}{2}\ln x - \frac{x^2}{4} - c_1(x-1)^2\right]_1^{k+1} + c_1 k \\
&= 2(k+1)\ln(k+1) - (k+1) - 4c_1\frac{k^2}{k+1} + \frac{4}{k+1} + c_1 k \\
&= 2(k+1)\ln(k+1) - \frac{(k+1)^2-4}{k+1} - 2\frac{c_1}{2}k\left(\frac{4k}{k+1}-1\right) \\
&< 2(k+1)\ln(k+1) - \frac{c_1}{2}k
\end{aligned} \tag{33}$$

よって、$n=k+1$ のときも成り立ち、(32) 式は任意の正の整数 n について成り立つ。すなわち、

$$T(n) \leq 2n\ln n - \frac{c_1}{2}(n-1) = \mathcal{O}(n\ln n)$$

より、$T(n)=\mathcal{O}(n\ln n)$ となる。